A Principled Technologies report: Hands-on testing. Real-world results.

A Principled Technologies report: Hands-on testing. Real-world results.

Up to 1.50x the rate of WordPress requests per second on 4vCPU VMs*

Up to 1.53x the rate of WordPress requests per second on 8vCPU VMs*

Up to 1.49x the rate of WordPress requests per second on 16vCPU VMs*

Handle WordPress web traffic with new Ddsv5 series VMs from Microsoft Azure featuring 3rd Generation Intel Xeon Scalable processors

The newer series outperformed previous-generation Ddsv4 series VMs featuring 2nd Generation Intel Xeon Scalable processors

If your business website runs on the cloud, the configurations of your chosen cloud VMs have a powerful effect on the performance your users and customers will see when interacting with your site. While newer hardware can often outperform older technology in a series, knowing the magnitude of the performance difference is important for businesses considering the VMs that will support their sites.

At Principled Technologies, we compared the WordPress request rate performance of two series of VMs from Microsoft Azure: newer Ddsv5 series VMs featuring 3rd Generation Intel Xeon Scalable processors and older Dds v4 series VMs featuring 2nd Generation Intel Xeon Scalable processors. Across three different VM sizes and configurations, the Ddsv5 series VMs processed a higher rate of requests per second than their previous-generation counterparts, suggesting that the new VMs could enable organizations to support a higher volume of website traffic and interactions.

*Ddsv5 series VMs featuring 3rd Gen Intel Xeon Scalable processors vs. Ddsv4 VMs featuring 2nd Gen Intel Xeon Scalable processors

Handle WordPress web traffic with new Ddsv5 series VMs from Microsoft Azure featuring 3rd Generation Intel Xeon Scalable processors

November 2021 (Revised)

The science behind the report:

Handle WordPress web traffic with new Ddsv5 series VMs from Microsoft Azure featuring 3rd Generation Intel Xeon Scalable processors

This document describes what we tested, how we tested, and what we found. To learn how these facts translate into real-world benefits, read the report Handle WordPress web traffic with new Ddsv5 series VMs from Microsoft Azure featuring 3rd Generation Intel Xeon Scalable processors.

We concluded our hands-on testing on October 8, 2021. During testing, we determined the appropriate hardware and software configurations and applied updates as they became available. The results in this report reflect configurations that we finalized on October 4, 2021 or earlier. Unavoidably, these configurations may not represent the latest versions available when this report appears.

Our results

To learn more about how we have calculated the wins in this report, go to http://facts.pt/calculating-and-highlighting-wins. Unless we state otherwise, we have followed the rules and principles we outline in that document.

Table 1: Results of our WordPress website transaction testing on Microsoft Azure VMs with 4, 8, and 16 vCPUs. We measured the average rate of requests per second (RPS) each VM achieved using five different TLS ciphers. In our main report, we cited results from the cipher shown in bold.

TLS cipher	Ddsv4 results (RPS)	Ddsv5 results (RPS)	Ddsv5 VM advantage
4vCPU results			
Benchmark-default cipher	157.5	205.2	1.30x
ECDHE-RSA-AES128-GCM-SHA256:ECDH:RSA:secp384r1	139.9	199.6	1.42x
ECDHE-RSA-AES128-GCM-SHA256:ECDH:RSA:X25519	157.3	206.5	1.31x
TLS_AES_256_GCM_SHA384:ECDH:ECDSA:secp384r1	**135.8**	**204.1**	**1.50x**
TLS_AES_256_GCM_SHA384:ECDH:ECDSA:X25519	165.5	211.0	1.27x

TLS cipher	**Ddsv4 results (RPS)**	**Ddsv5 results (RPS)**	Ddsv5 VM advantage
8 vCPU results			
Benchmark Default	318.2	428.2	1.34x
ECDHE-RSA-AES128-GCM-SHA256:ECDH:RSA:secp384r1	282.9	416.5	1.47x
ECDHE-RSA-AES128-GCM-SHA256:ECDH:RSA:X25519	320.8	427.8	1.33x
TLS_AES_256_GCM_SHA384:ECDH:ECDSA:secp384r1	**271.9**	**417.0**	**1.53x**
TLS_AES_256_GCM_SHA384:ECDH:ECDSA:X25519	329.7	430.2	1.30x
16 vCPU results			
Benchmark Default	646.9	845.3	1.30x
ECDHE-RSA-AES128-GCM-SHA256:ECDH:RSA:secp384r1	572.4	820.7	1.43x
ECDHE-RSA-AES128-GCM-SHA256:ECDH:RSA:X25519	652.0	848.1	1.30x
TLS_AES_256_GCM_SHA384:ECDH:ECDSA:secp384r1	**550.6**	**824.2**	**1.49x**
TLS_AES_256_GCM_SHA384:ECDH:ECDSA:X25519	665.6	851.8	1.27x

System configuration information

Table 2: Detailed information on the Microsoft Azure Ddsv5 VMs we tested.

System configuration information	D4ds_v5 VM	D8ds_v5 VM	D16ds_v5 VM
Tested by	Principled Technologies	Principled Technologies	Principled Technologies
Test date	10/7/2021	10/7/2021	10/7/2021
CSP / Region	East US 2	East US 2	East US 2
Workload & version	oss-performance (Intel modified)	oss-performance (Intel modified)	oss-performance (Intel modified)
Iterations and result choice	3 runs, median	3 runs, median	3 runs, median
Server platform	D4ds_v5	D8ds_v5	D16ds_v5
BIOS name and version	Microsoft Corporation Hyper-V UEFI Release v4.1, 10/27/2020	Microsoft Corporation Hyper-V UEFI Release v4.1, 10/27/2020	Microsoft Corporation Hyper-V UEFI Release v4.1, 10/27/2020
Operating system name and version/build number	Bitnami wordpress-intel 5.8.1-4, Debian 4.19.194-3	Bitnami wordpress-intel 5.8.1-4, Debian 4.19.194-3	UBitnami wordpress-intel 5.8.1-4, Debian 4.19.194-3
Date of last OS updates/ patches applied	10/6/2021	10/6/2021	10/6/2021
Processor			
Number of processors	4	8	16
Vendor and model	Intel® Xeon® Platinum 8370C	Intel Xeon Platinum 8370C	Intel Xeon Platinum 8370C
Core count (per processor)	2	4	8
Core frequency (GHz)	2.8	2.8	2.8
Stepping	6	6	6
Hyper-Threading	Yes	Yes	Yes
Turbo	Yes	Yes	Yes
Number of vCPU per VM	4	8	16
Memory module(s)			
Total memory in system (GB)	16	32	64
NVMe memory present?	No	No	No
Total memory (DDR+NVMe RAM)	16	32	64
General hardware			
Storage: NW or Direct Att / Instance	NW Att	NW Att	NW Att
Network BW / Instance	N/A	N/A	N/A
Storage BW / Instance	N/A	N/A	N/A

System configuration information	D4ds_v5 VM	D8ds_v5 VM	D16ds_v5 VM
Local storage			
OS			
Number of drives	1	1	1
Drive size (GB)	30	30	30
Drive information (speed, interface, type)	Standard SSD	Standard SSD	Standard SSD
Temporary drive			
Number of drives	0	0	0
Drive size (GB)	N/A	N/A	N/A
Network adapter			
Vendor and model	Microsoft Hyper-V Network Adapter	Microsoft Hyper-V Network Adapter	Microsoft Hyper-V Network Adapter
Number and type of ports	1x 100Gb	1x 100Gb	1x 100Gb

Table 3: Detailed information on the Microsoft Azure Ddsv4 VMs we tested.

System configuration information	D4ds_v4 VM	D8ds_v4 VM	D16ds_v4 VM
Tested by	Principled Technologies	Principled Technologies	Principled Technologies
Test date	10/7/2021	10/7/2021	10/7/2021
CSP / Region	East US 2	East US 2	East US 2
Workload & version	oss-performance (Intel modified)	oss-performance (Intel modified)	oss-performance (Intel modified)
Iterations and result choice	3 runs, median	3 runs, median	3 runs, median
Server platform	D4ds_v4	D8ds_v4	D16ds_v4
BIOS name and version	Microsoft Corporation Hyper-V UEFI Release v4.1, 10/27/2020	Microsoft Corporation Hyper-V UEFI Release v4.1, 10/27/2020	Microsoft Corporation Hyper-V UEFI Release v4.1, 10/27/2020
Operating system name and version/build number	Bitnami wordpress-intel 5.6.1-4, Debian 4.19.194-3	Bitnami wordpress-intel 5.6.1-4, Debian 4.19.194-3	Bitnami wordpress-intel 5.6.1-4, Debian 4.19.194-3
Date of last OS updates/ patches applied	10/6/2021	10/6/2021	10/6/2021
Processor			
Number of processors	4	8	16
Vendor and model	Intel Xeon Platinum 8272CL	Intel Xeon Platinum 8272CL	Intel Xeon Platinum 8272CL
Core count (per processor)	2	4	8
Core frequency (GHz)	2.60	2.60	2.60
Stepping	7	7	7
Hyper-Threading	Yes	Yes	Yes
Turbo	Yes	Yes	Yes
Number of vCPU per VM	4	8	16

System configuration information	D4ds_v4 VM	D8ds_v4 VM	D16ds_v4 VM
Memory module(s)			
Total memory in system (GB)	16	32	64
NVMe memory present?	No	No	No
Total memory (DDR+NVMe RAM)	16	32	64
General hardware			
Storage: NW or Direct Att / Instance	NW Att	NW Att	NW Att
Network BW / Instance	N/A	N/A	N/A
Storage BW / Instance	N/A	N/A	N/A
Local storage			
OS			
Number of drives	1	1	1
Drive size (GB)	30	30	30
Drive information (speed, interface, type)	Standard SSD	Standard SSD	Standard SSD
Temporary drive			
Number of drives	0	0	0
Drive size (GB)	N/A	N/A	N/A
Network adapter			
Vendor and model	Microsoft Hyper-V Network Adapter	Microsoft Hyper-V Network Adapter	Microsoft Hyper-V Network Adapter
Number and type of ports	1x 100Gb	1x 100Gb	1x 100Gb

Table 4: Detailed information on the client system we used.

System configuration information	8vCPU client VM
Tested by	Principled Technologies
Test date	10/7/2021
CSP / Region	East US 2
Workload & version	oss-performance (Intel modified)
Iterations and result choice	3 runs, median
Server platform	D8ds_v4
BIOS name and version	Microsoft Corporation Hyper-V UEFI Release v4.1, 10/27/2020
Operating system name and version/build number	Ubuntu Server 20.04 LTS, Debian 5.8.0-1042-azure
Date of last OS updates/patches applied	10/6/2021

System configuration information	8vCPU client VM
Processor	
Number of processors	8
Vendor and model	Intel Xeon Platinum 8272CL
Core count (per processor)	4
Core frequency (GHz)	2.60
Stepping	7
Hyper-Threading	Yes
Turbo	Yes
Number of vCPU per VM	8
Memory module(s)	
Total memory in system (GB)	32
NVMe memory present?	No
Total memory (DDR+NVMe RAM)	32
General hardware	
Storage: NW or Direct Att / Instance	NW Att
Network BW / Instance	N/A
Storage BW / Instance	N/A
Local storage	
OS	
Number of drives	1
Drive size (GB)	30
Drive information (speed, interface, type)	Standard SSD
Temporary drive	
Number of drives	0
Drive size (GB)	N/A
Network adapter	
Vendor and model	Microsoft Hyper-V Network Adapter
Number and type of ports	1x 100Gb

How we tested

Testing overview

We created our WordPress VMs using the "WordPress for Intel packaged by Bitnami - Gen2" in the Azure image library. This image includes NGINX, PHP, MariaDB, and Wordpress pre-installed along with the latest Intel(R) QuickAssist Technology (QAT) enhancements for increased SSL performance on the latest Intel v5 instances (Ice Lake). We used the same size database and workload for all instances. Due to the different number of vCPUs between instances some tuning settings differ among the three pairs. However, within each pair, we kept everything as identical as possible outside of the instance types themselves. See below for the steps we followed as well as tables to show the VM configurations and settings. The database is small and easily fits in RAM so we didn't need to worry about fluctuations in disk performance, although we used Standard SSD storage for all instances just to ensure there was no delay loading the database into memory. Additionally create D8ds_v4 instance for the benchmark client (named wordpress-client).

Creating the client VM using Ubuntu Server 20.04 LTS

This section contains the steps we took to create our client instance.

Create the client VM

1. Log into the Azure Portal and navigate to the Virtual Machines service.
2. Click Add to open the Add VM wizard.
3. On the Basics tab, set the following:
    a. Choose your Subscription from the dropdown menu.
    b. Choose your Resource group from the dropdown menu.
    c. Name the Virtual Machine.
    d. Choose your Region from the dropdown menu.
    e. Leave the Availability options set to No infrastructure redundancy required.
    f. Click Browse all public and private images
    g. In the Search field, enter Ubuntu Server 20.04 LTS
    h. Select "Ubuntu Server 20.04 LTS - Gen2" from the list of results.
    i. Leave Azure Spot instance set to No.
    j. Select the instance size you wish to use, we used D8ds_v4.
    k. Leave the Authentication type set to SSH public key.
    l. Either choose a new Username or leave the default.
    m. Choose Generate new key pair for the SSH public key source.
    n. Enter a name for the Key pair name.
    o. Leave Public inbound ports set to Allow selected ports.
    p. For Select inbound ports, choose SSH (22).
4. On the Disks tab, set the following:
    a. For the OS disk type, choose Standard SSD from the dropdown menu.
    b. Leave the default Encryption type.
5. On the Networking tab, set the following:
    a. Choose your Virtual network from the dropdown menu.
    b. Choose Create new to create a new Public IP.
    c. Leave the rest of the settings at defaults.
6. On the Management tab, set the following:
    a. Choose your Diagnostics storage account from the dropdown menu.
    b. Leave the rest set to defaults.
7. On the Advanced tab, leave all defaults
8. On the Tags tab, add any tags you wish to use.
9. On the Review + create tab, review your settings, and click Create

Configure Ubuntu Server 20.04 LTS

Log in as azureuser user using the SSH key generated during Azure instance creation. Example:

```
ssh -i wordpress_key.pem azureuser@<INSTANCE_PUBLIC_IP_ADDRESS>
```

Install the latest update packages and reboot the VM.

```
sudo apt-get update
    sudo apt-get upgrade -y
    sudo reboot
```

Set the timezone on the VM.

```
sudo timedatectl set-timezone America/New_York
```

Install additional tools:

```
sudo apt-get install -y nmon ksh numactl virt-what jq spectre-meltdown-checker siege git
    wget sysstat lshw
```

Download benchmark client files and prerequisites

Download latest siege client URL file:

```
wget https://raw.githubusercontent.com/intel/Updates-for-OSS-Performance/main/targets/wordpress/
    WordpressTarget_v5.urls
```

Install nmonchart as a prerequisite to running the benchmark scripts:

```
wget https://raw.githubusercontent.com/aguther/nmonchart/master/nmonchart
    chmod +x nmonchart
```

Create the VMs under test

In this section we list the steps required to create a VM from the "WordPress for Intel packaged by Bitnami - Gen2" in the Azure image library. See tables 2 and 3 for the list of VMs we used and follow the steps six times using the proper specifications for the VM you wish to create For our testing, we used the East US Region and Availability Zone 2.

Create the VMs from the WordPress for Intel image

1. Log into the Azure Portal and navigate to the Virtual Machines service.
2. Click Add to open the Add VM wizard.
3. On the Basics tab, set the following:
    a. Choose your Subscription from the dropdown menu.
    b. Choose your Resource group from the dropdown menu.
    c. Name the Virtual Machine.
    d. Choose your Region from the dropdown menu.
    e. Leave the Availability options set to No infrastructure redundancy required.
    f. Click Browse all public and private images
    g. In the Search field, enter WordPress for Intel
    h. Select " WordPress for Intel packaged by Bitnami - Gen2" from the list of results.
    i. Leave Azure Spot instance set to No.
    j. Select the instance size you wish to use.
    k. Leave the Authentication type set to SSH public key.
    l. Either choose a new Username or leave the default.
    m. Choose Generate new key pair for the SSH public key source.
    n. Enter a name for the Key pair name.
    o. Leave Public inbound ports set to Allow selected ports.
    p. For Select inbound ports, choose SSH (22).

4. On the Disks tab, set the following:
    a. For the OS disk type, choose Standard SSD from the dropdown menu.
    b. Leave the default Encryption type.
5. On the Networking tab, set the following:
    a. Choose your Virtual network from the dropdown menu.
    b. Choose Create new to create a new Public IP.
    c. Leave the rest of the settings at defaults.
6. On the Management tab, set the following:
    a. Choose your Diagnostics storage account from the dropdown menu.
    b. Leave the rest set to defaults.
7. On the Advanced tab, leave all defaults
8. On the Tags tab, add any tags you wish to use.
9. On the Review + create tab, review your settings, and click Create

Configure the WordPress VMs

1. Log in as azureuser user using the SSH key generated during Azure instance creation. Example:

```
ssh -i wordpress_key.pem azureuser@<INSTANCE_PUBLIC_IP_ADDRESS>
```

2. Install the latest update packages and reboot the VM.

```
sudo apt-get update
    sudo apt-get upgrade -y
    sudo reboot
```

3. Set the timezone on the VM.

```
sudo timedatectl set-timezone America/New_York
```

4. Install additional tools:

```
sudo apt-get install -y nmon ksh numactl virt-what jq spectre-meltdown-checker siege git wget sysstat
    curl lsb-release lshw
```

Load WordPress Database:

1. Stop all WordPress services and restart MariaDB:

```
sudo /opt/bitnami/ctlscript.sh stop
    sudo /opt/bitnami/ctlscript.sh start mariadb
```

2. Download and extract the WordPress database backup for the benchmark:

```
wget https://github.com/intel/Updates-for-OSS-Performance/raw/main/targets/wordpress/dbdump_v5.sql.gz
gunzip dbdump_v5.sql
```

3. Locate the generated MariaDB password from the image:

```
sudo cat /home/bitnami/bitnami_credentials
```

4. Load the WordPress becnhmark database backup into MariaDB using the password obtained in the previous step:

```
cat dbdump_v5.sql | mysql -u root --password=<MARIADB_PASSWORD> bitnami_wordpress
```

5. Restart all WordPress services:

```
sudo /opt/bitnami/ctlscript.sh restart
```

Disable QAT enhancements on v4 instances

1. Since the Intel QAT enhancements are not supported on instances older than v5, you will need to disable them for the v4 instances. Remove the following lines from the /opt/bitnami/nginx/conf/nginx.conf file to disable them:

```
load_module modules/ngx_ssl_engine_qat_module.so;
    ssl_engine {
        use_engine qatengine;
        default_algorithms RSA,EC,DH,PKEY_CRYPTO;
        qat_engine {
            qat_offload_mode async;
            qat_notify_mode poll;
            qat_poll_mode heuristic;
            qat_shutting_down_release on;
        }
    }
```

Generate and exchange SSH keys

1. On the wordpress-client VM, run the following command to create a new SSH key pair:

```
ssh-keygen
```

2. Press enter four times to save the key pair to the default location with no password.
3. Once the key pair has been generated, run the following command to obtain your new public SSH key:

```
cat .ssh/id_rsa.pub
```

4. Copy the output key to your clipboard.
5. Log into the VM under test.
6. Open the authorized_key file with the following command, and copy the public SSH key from the VM under test into the file:

```
sudo vim .ssh/authorized_keys
```

7. Repeat steps 1-6 for the other VMs under test, and the wordpress-client VM, copying the public SSH key to the authorized_keys file.

Generate additional OpenSSL certificates

On the VM under test, run the following commands to create additional SSL certificates used in the benchmark:

```
CERT_DIR=/opt/bitnami/nginx/conf/bitnami
openssl req -newkey rsa:2048 -nodes -keyout $CERT_DIR/certs/rsa.key -x509 -days 365 -out $CERT_DIR/
    certs/rsa.crt -subj "/C=/ST=/L=/O=/OU=/CN=$(hostname)" 2> /dev/null
openssl ecparam -genkey -out $CERT_DIR/certs/dsa-prime256v1.key -name prime256v1
openssl req -x509 -new -key $CERT_DIR/certs/dsa-prime256v1.key -out $CERT_DIR/certs/dsa-prime256v1.
    crt -subj "/C=/ST=/L=/O=/OU=/CN=$(hostname)" 2> /dev/null
openssl ecparam -genkey -out $CERT_DIR/certs/dsa-secp384r1.key -name secp384r1
openssl req -x509 -new -key $CERT_DIR/certs/dsa-secp384r1.key -out $CERT_DIR/certs/dsa-secp384r1.crt
    -subj "/C=/ST=/L=/O=/OU=/CN=$(hostname)" 2> /dev/null
```

Run the tests

In this section, we list the steps to run the oss-performance benchmark on the VMs under test. The benchmark is started form the client VM using a script that automates the entire process.

1. Run the benchmark script substituting the hostname of the VM under test:

```
./run_test_image.sh <HOSTNAME_OF_VM_UNDER_TEST>
```

2. Results are automatically saved in the results/ subdirectory of the client VM.
3. Repeat all tests 3 times for all VM instance types.

Scripts

Ice Lake mitigations

```
Vulnerability Itlb multihit: Not affected
Vulnerability L1tf: Not affected
Vulnerability Mds: Not affected
Vulnerability Meltdown: Not affected
Vulnerability Spec store bypass: Vulnerable
Vulnerability Spectre v1: Mitigation; usercopy/swapgs barriers and __user pointer sanitization
Vulnerability Spectre v2: Mitigation; Full generic retpoline, STIBP disabled, RSB filling
Vulnerability Srbds: Not affected
Vulnerability Tsx async abort: Vulnerable: Clear CPU buffers attempted, no microcode; SMT
    Host state unknown
```

Cascade Lake mitigations

```
Vulnerability Itlb multihit:     KVM: Mitigation: Split huge pages
Vulnerability L1tf:              Not affected
Vulnerability Mds:               Not affected
Vulnerability Meltdown:          Not affected
Vulnerability Spec store bypass: Vulnerable
Vulnerability Spectre v1:        Mitigation; usercopy/swapgs barriers and __user pointer sanitization
Vulnerability Spectre v2:        Mitigation; Full generic retpoline, STIBP disabled, RSB filling
Vulnerability Srbds:             Not affected
Vulnerability Tsx async abort:   Vulnerable: Clear CPU buffers attempted, no microcode; SMT
    Host state unknown
```

Read the report at http://facts.pt/Hz8dkm9 ▶

This project was commissioned by Intel.

Facts matter.®

DISCLAIMER OF WARRANTIES; LIMITATION OF LIABILITY:
Principled Technologies, Inc. has made reasonable efforts to ensure the accuracy and validity of its testing, however, Principled Technologies, Inc. specifically disclaims any warranty, expressed or implied, relating to the test results and analysis, their accuracy, completeness or quality, including any implied warranty of fitness for any particular purpose. All persons or entities relying on the results of any testing do so at their own risk, and agree that Principled Technologies, Inc., its employees and its subcontractors shall have no liability whatsoever from any claim of loss or damage on account of any alleged error or defect in any testing procedure or result.

In no event shall Principled Technologies, Inc. be liable for indirect, special, incidental, or consequential damages in connection with its testing, even if advised of the possibility of such damages. In no event shall Principled Technologies, Inc.'s liability, including for direct damages, exceed the amounts paid in connection with Principled Technologies, Inc.'s testing. Customer's sole and exclusive remedies are as set forth herein.